Algorithms, Data Structures and Complexity (DT505G)

Exam Example

Teacher:	Federico Pecora (ext. 3319)
Date:	—
Time:	—
Material:	No material allowed except dictionary
No. exercises:	6
Total points:	70 (42 required to pass)
No. pages:	7 (excluding this page)

- Start each exercise on a **new page**
- Only write **on one side** of the page
- When handing in, please organize pages in the **correct order**
- **Do not write your name** anywhere, use your **exam code** only
- **Do not** use a **red pen**
- If a question or exercise is unclear, make reasonable assumptions and **write them down**
- Please use **clear handwriting**
- Answers must be given in **English or Swedish**
- Please use **short and complete sentences**

Exercise 1 (20 points)

Answer the following statements with either True or False. Correct answers give 1 point. If the statement ends with the sentence "(explain your answer)", you can get an additional point by providing a correct and more detailed answer to the corresponding question.

S1 The worst-case computational complexity of Breadth-First Search (BFS) is asymptotically higher than the worst-case computational complexity of Depth-First Search (DFS) (explain your answer).

S2 Given an undirected graph $G = (V, E)$ and a source node $s \in V$, Breadth-First Search $\text{BFS}(G, s)$ computes the shortest path $\delta(s, v)$ between s and all nodes $v \in V$ (explain your answer).

S3 Given a directed graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$, the algorithm $\textsc{Bellman-Ford}(G, w, s)$ returns $\textsc{False}$ if and only if there exists a cycle p such that $\sum_{(i,j)\in p} w_{ij} < 0$.

S4 Given a directed acyclic graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$ and a topological sort of the nodes in V, computing the shortest path between a source node s and all other nodes requires at least $|V|^2$ calls to $\textsc{Relax}(u, v, w)$ (explain your answer).

S5 Dijkstra's algorithm computes the minimum distances between a given source and all nodes in a graph (explain your answer).

S6 The Floyd-Warshall algorithm computes the minimum distances between all pairs of nodes in a weighted graph.

S7 Given a dynamic set with n elements, the minimum depth of a Binary Search Tree representation of the set is $n \log n$ (explain your answer).

S8 All known algorithms for sorting a dynamic set have exponential worst-case complexity (explain your answer).

S9 If $f(n) = n^3/2 - 2n$ and $g(n) = 3n$, then $f(n) = O(g(n))$ (explain your answer).

S10 A stack is a dynamic set which follows the First-In-First-Out (FIFO) policy.

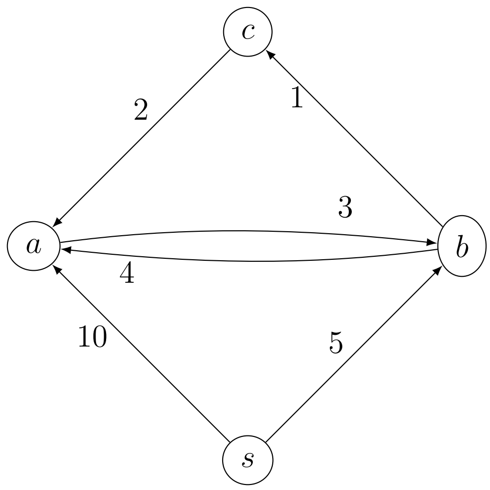
S11 The worst-case running time of the Insersion sort algorithm for sorting dynamic sets is the same as the worst-case running time of the Quicksort algorithm (explain your answer).

S12 If you find a polynomial-time algorithm for solving a problem that is NP-Complete, then you can use this algorithm to solve all problems in NP in polynomial time.

Exercise 2 (10 points)

Describe Dijkstra's algorithm, answering in particular the following questions: what problem does the algorithm solve? Under which conditions is the algorithm correct (which assumptions does it make on the input graph)? What is the worst-case computational complexity of the algorithm?

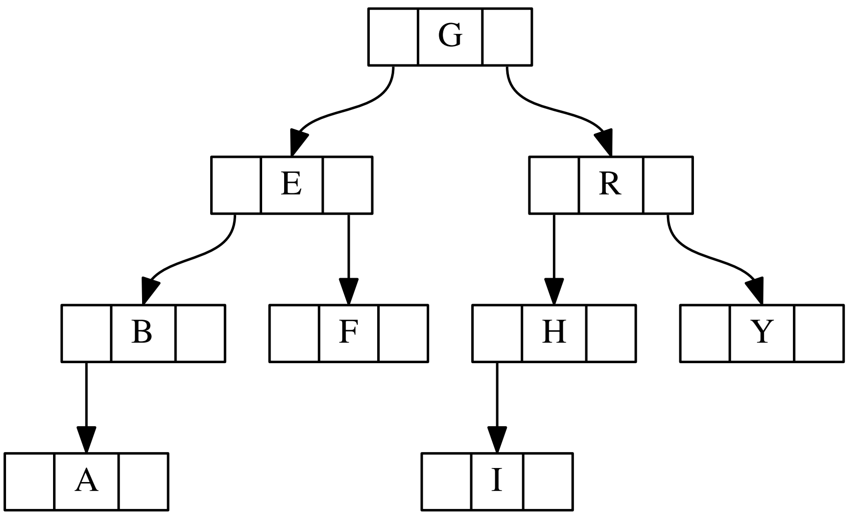
Illustrate how the algorithm works with the following directed weighted graph, where s is the source node.



Exercise 3 (10 points)

Illustrate the Binary Search Tree (BT) data structure. Describe the algorithms for inserting a new element into a BT, finding the maximum element in a BT, finding the successor (or predecessor) of an element in a BT. What is the computational cost of these operations? Why is it important that BTs are balanced?

Illustrate how the BT below is modified after performing the following two operations in order: insert a new element with key "D"; remove the element with key "H".



Exercise 4 (10 points)

You are given graph $G = (V, E)$ with $|V| = n$ vertices and weight function $w : E \to \mathbb{R}$. You are also given a $n \times n$ matrix

$$W^{(0)} = \left(w_{ij}^{(0)}\right),$$

where $w_{ij}^{(0)}$ is the value in the i-th row and j-th column. The matrix represents the weights of the edges in G, that is,

$$w_{ij}^{(0)} = \begin{cases} w(i,j), & \text{if } (i,j) \in E, \\ \infty, & \text{otherwise.} \end{cases}$$

The pseudo-code below is a dynamic programming algorithm which takes as input the matrix $W^{(0)}$ (you have studied this algorithm in the course).

Mystery-Algorithm($W^{(0)}$)

$n = W.rows$
for $k = 1$ **to** n
  let $W^{(k)} = \left(w_{ij}^{(k)}\right)$ be a new $n \times n$ matrix
  for $i = 1$ **to** n
    for $j = 1$ **to** n
      $w_{ij}^{(k)} = \min(w_{ij}^{(k-1)}, w_{ik}^{(k-1)} + w_{kj}^{(k-1)})$
return $W^{(n)}$

What does the algorithm do? How does the algorithm modify the input matrix? What is the computational complexity of the algorithm?

Exercise 5 (10 points)

Illustrate the Random Access Machine compuational model. Address the following points:

- Why do we need a computational model to study algorithms?
- What kinds of instructions do we assume that a RAM can perform?
- Can you give one or more examples of instruction that we assume a RAM *cannot* execute?
- What amount of time is required to perform these instructions?
- Can instructions be executed concurrently?
- What datatypes do we assume that we can represent?
- Why do we assume that the size of the datatypes is bounded?

Exercise 6 (10 points)

Illustrate two data structures for representing directed graphs. Compare the memory usage of the two data structures. Use the graph below as an example for explaining the representations.